Table of Contents

During the design phase of JVx, particular attention was paid to **openness**. We have developed a **complete package** that covers all requirements of **enterprise applications** out of the box without neglecting the extendability.

No new standards were created, as the selection of standards is already big enough. JVx sets off with **well-known techniques** and **new ideas**.

Based on a multitier architecture, JVx consists of the following parts:

The tiers have following missions:

- **Client Tier**
  The client tier allows all client technologies (rich, fat, thin). For us, this includes, among others, Java Swing, SWT, QT Jambi, Silverlight, .NET Framework, and, of course, also modern Javascript libraries and techniques such as jQuery, extJS, qooxdoo, GWT, echo3, and many more.
  The client tier has no direct access to data or to the data tier, and can only communicate with the enterprise tier.
- **Enterprise Tier**
  The centre piece of JVx is anchored in the enterprise tier. A communication server takes in the requests from the llient tier, processes them, and mediates the prepared result. The server itself is an exchangeable and extendable component and makes use of further components for session administration, direct object access, and access control.
  For database access, a Persistence API is made available so as to enable a technology-independent access to the data tier.
- **Data Tier**
  The data tier can be any data container, such as relation database systems (e.g., Oracle, DB2, MySql) or also special data formats (e.g., XML, XLS).

More information can be found in the Package Overview.

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/jvx/system_architecture

Last update: **2020/08/05 12:03**